Electronic Journal of Graph Theory and Applications **8** (1) (2020), 113–122

Electronic Journal of
Graph Theory and Applications

Determining the robustness of an interdependent network with a hypergraph model

Gholam Hasan Shirdel, Ameneh Mortezaee

Department of Mathematics, University of Qom, Qom, I. R. Iran

shirdel81math@gmail.com, A.Mortezaee@stu.qom.ac.ir

Abstract

The world is included of various entities and complex interdependencies between them that can be appeared in multi-layered networks. It may be the acting of some of these entities depends on the acting of the others such that the failure in one entity may cause failures in a number of others. In this paper we try to model these complex interdependencies in a interdependent network with a directed hypergraph model and then we propose an algorithm to determine minimum number of failure for total failure in the power grid and communication network as a special interdependent network.

Keywords: interdependent network, power grid and communication network, hypergraph, robustness
Mathematics Subject Classification: 05C65, 90B18
DOI: 10.5614/ejgta.2020.8.1.8

1. Introduction

In the last few years infrastructure networks that are included of several types of entities as nodes and several types of relationships as edges, have been increasing development. In many cases theses networks have interdependencies and operation of one entity directly effects on the operation of another such that the failure on one entity leads to failure on one or more entities. Such networks have been known as interdependent multi-layered networks that can be named Power Grid and supporting Control and Communication Network (PG and CCN) as an instance of them. This network has two layers. first layer's elements consist of power stations and substations, are

Received: 23 January 2017, Revised: 13 June 2019, Accepted: 27 October 2019.

controlled by CCN while second layer's elements consist of routers can not operate without electric power generated in the first layer. Therefore failure in any element may cause disorderliness of other elements and so failure cascades throughout the network.

Because of existing a lot of complex interdependencies, prediction of cascading failures in such networks are very difficult. Recently significant efforts have been made in studying and analyzing interdependencies in multi-layered networks [4, 5, 7, 8, 9, 10, 12, 13, 15, 16]. Papers by Rosato [12] and Rinaldi [11] described the concept of interdependency but non of them proposed a mathematical model for deliberation of such networks. In 2010 Buldyrev et.al presented a model for studying the robustness of interdependent random networks and then analyzing cascading random failure in this network [5]. Later Parandehgheibi et al. in 2013 produced a mathematical graph based model for determining robustness of PG and CCN in which by using cycles of the network designated minimum number of nodes whose removals were led to total failure [10]. However finding all cycles of a network is very difficult, therefore this method may be impracticable in large networks.

It's essential to note that in all above works, interdependencies have been considered only in special cases and any model have not been presented that can capture all possible cases. For example in highly cited paper [5] authors proposed model in which every node depended on one and only one node of other network and later in [7] the same authors argue that this interdependencies may not be valid in real world.

However in addition to the number of interdependencies, the types of them also may be different. we are consider two types of interdependencies:

- disjunctive dependency
- conjunctive dependency

For more details let when we say a_i is alive as long as either b_j or b_k is alive, in other word we write $b_j + b_k \to a_i$, we have a disjunctive dependency and when we say a_i is alive as long as both b_j and b_k are alive, in other word we write $b_j b_k \to a_i$, we have a conjunctive dependency. This latter dependency has not been considered in many of presented models. In [12] Arunabha Sen et al. presented a model for capturing such dependency in which by using linear programming model solved some problems in the field of robustness. However some drawbacks are seen in some conditions of this LP.

We use a hypergraph model in our method that simply captures all information about these dependencies and then by using of the number of closed walks passing every node, propose criteria for robustness of the network, i.e. a quantity that shows the network's ability in transmission the flow. hypergraphs are a generalization of graphs in which edges, known as hyperedges, can connect sets of more than two nodes [2], [3].

In this paper we study PG and CCN as a special case of interdependent two-layered network. However these analysis can be extended on every interdependent multi-layered network. In the interdependent multi-layered network, the nodes of two or more single network are adjacent to each other by interdependency edges. Hence if $\mathcal{N} = (\mathcal{V}, \mathcal{A})$ is an interdependent n-layered network then we will have: $\mathcal{V} = \mathcal{V}_1 \cup \cdots \cup \mathcal{V}_n$ in which $\mathcal{V}_i$ is the node set of ith layer and $\mathcal{A}$ is the set of interdependency edges. Since our focus in this paper is on interdependency edges, so for simplicity

www.ejgta.org

we ignore intra edges in each layer. Now we must consider all interdependencies of each node on other nodes or other edges. for example if e_k^{ij} is the kth interdependency edge from ith layer to jth layer between v_t^i and v_r^j i.e $e_k^{ij} = (v_t^i, v_r^j)$, then v_r^j is dependent on existence of v_t^i and on existence of e_k^{ij}. Therefore we can consider a directed hyperedge as $(\{v_t^i, e_k^{ij}\}, v_r^j)$. it can be shown in figure (1).

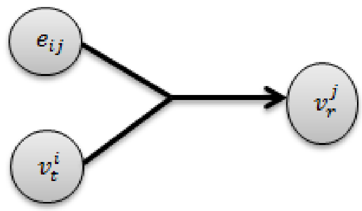

Figure 1. A conjunctive interdependency

In addition, v_l^i may depend on two or more nodes conjunctively. Also in this case we can consider a directed hyperedge in which the tail consists of those nodes and corresponding interdependencies edges and its head equals v_l^i.

In general, first we consider all interdependencies of a node on nodes or edges in the multi-layered network $\mathcal{N}$ and then we define a directed hypergraph $\mathcal{H} = (\mathcal{V}_\mathcal{H}, \mathcal{A}_\mathcal{H}$ in which $\mathcal{V}_\mathcal{H}$ consists of all entities in $\mathcal{N}$ and $\mathcal{A}_\mathcal{H}$ consists of all interdependencies of nodes on other nodes and edges as we say above.

Our goal is finding minimum number of failures that lead to total failure. Before proposing the algorithm, we must talk about cascading failures and for simplification we study this concept in an interdependent two-layered network, PG and CCN. However as said before, we can generalize it easily.

The rest of this paper is divided as follows: In the next section PG and CCN will be introduced. In section 3 a directed hypergraph model will be proposed and then in section 4, we will present an algorithm for determining the robustness of PG and CCN. In section 5 our method will be evaluated using an example and finally section 6 will be conclusion.

2. Power Grid and supporting Control and Communication Network

In this section we start by introduction of PG and CCN and then analyze the concept of cascading failures in it.

A PG and CCN is a interdependent two-layered network. first layer consists of a generator G and a number of substations s_i for $i = 1, \cdots, n$ that connect to each other by power lines. We assume that G sends power to each s_i separately. Therefore we can ignore relationships between s_i's in first layer. Second layer consists of a central controller C and a number of routers r_j for $j = 1, \cdots, m$ that connect to each other by communication lines. Similarly we assume that every router is controlled by C separately. Then we can ignore relationships between r_j's in second layer. Then we can consider this network as $\mathcal{N} = (\mathcal{S} \cup \mathcal{R}, \mathcal{A})$ where $\mathcal{S}$ is the set of substations and $\mathcal{R}$ is the set of routers and finally $\mathcal{A}$ is the set of all interdependency edges.

www.ejgta.org

Every router is alive if it receives power from at least one alive substation and every substation is alive if it receives control signal from at least one alive router. Therefore we have a set of interdependencies edges between these layers. Now as for these interdependencies, every failure in PG, for example in s_i, may cause failure in all routers that receive power from only s_i. Similarly every failure in CCN may cause failure or failures in PG. Therefore failures in one layer may cascade throughout the system alternatively and fail total of it.

In this paper we want to find minimum number of failures that causes total failure in the system. The case of disjunctive interdependencies has been solved in [10]. Now if interdependencies are also conjunctive, how can we solve this problem? In other word, if we have $s_i s_j s_l \to r_k$ how can represent this interdependency? Also if we have $r_j \to s_i$ and we want to represent interdependency of s_i on r_j and on $e = (r_j, s_i)$ simultaneously, what model must we use? We use a hypergraph model to represent all kinds of interdependencies. In the next section we introduce this hypergraph model.

3. A Directed Bipartite Hypergraph Model

We start with introduction a directed bipartite hypergraph.

Definition 1. A directed bipartite hypergraph is $\mathcal{H} = (\mathcal{V}_1 \cup \mathcal{V}_2, \mathcal{E}_\mathcal{H})$ in which $\mathcal{V}_\mathcal{H} = \mathcal{V}_1 \cup \mathcal{V}_2$ is the set of vertices and $\mathcal{V}_1 \cap \mathcal{V}_2 = \phi$ and $\mathcal{E}_\mathcal{H}$ is the set of directed hyperedges and we have:

$$\mathcal{E}_\mathcal{H} = \{ \ \mathcal{E} = (\mathcal{E}_1, \mathcal{E}_2) \mid \mathcal{E}_1 \subseteq \mathcal{V}_1 \ , \ \mathcal{E}_2 \subseteq \mathcal{V}_2 \ \}$$

where $\mathcal{E}_1$ is the tail of $\mathcal{E}$ and $\mathcal{E}_2$ is its head.

As we said before, we can model PG and CCN with all kinds of interdependencies between them by a directed bipartite hypergraph. The set of vertices of this hypergraph consists of all entities in PG and CCN i.e. substations and routers and interdependency edges. In other word we have $\mathcal{V}_\mathcal{H} = \mathcal{S} \cup \mathcal{R} \cup \mathcal{A}$. For its hyperedges we have:

$$\begin{array}{ccc} (\mathcal{E}_1, r_j) \in \mathcal{E}_\mathcal{H} & \iff & \mathcal{E}_1 \longrightarrow r_j \\ (\mathcal{E}'_1, s_i) \in \mathcal{E}_\mathcal{H} & \iff & \mathcal{E}'_1 \longrightarrow s_i \end{array}$$

where $\mathcal{E}_1 \subset \mathcal{S} \cup \mathcal{A}$ and $\mathcal{E}'_1 \ \subset \ \mathcal{R} \cup \mathcal{A}$.

Now we study the effect of cascading failures in this model. As we said before, a failure in s_i or r_j or a failure in relationship between them may cause cascading failures throughout the PG and CCN and this process may make total failure. Now we have an important question: what is the minimum number of failures that leads to total failure?

We consider it as a metric that measures the robustness of the PG and CCN. In the following we show that this metric is equivalent with minimum number of nodes whose removals from the hypergraph model hit all of its cycles. In order to do so, we start with the following definitions in directed hypergraph. Similar definitions about undirected unipartite hypergraph have been mentioned in [6].

Definition 2. Let $\mathcal{H} = (\mathcal{V}_\mathcal{H}, \mathcal{A}_\mathcal{H})$ be a directed hypergraph. A walk of length k in $\mathcal{H}$ is defined as a sequence of (not necessarily distinct) vertices and hyperedges $(\mathcal{V}_1, \mathcal{E}_1, \mathcal{V}_2, \cdots, \mathcal{V}_k, \mathcal{E}_k, \mathcal{V}_{k+1})$ such that for each $i = 1, \cdots, k$ if $\mathcal{E}_i = (\mathcal{E}_1^i, \mathcal{E}_2^i)$ then $\mathcal{V}_i \in \mathcal{E}_1^i$ and $\mathcal{V}_{i+1} \in \mathcal{E}_2^i$.

Definition 3. The walk $W = (\mathcal{V}_1, \mathcal{E}_1, \mathcal{V}_2, \cdots, \mathcal{V}_k, \mathcal{E}_k, \mathcal{V}_{k+1})$ is called a closed walk if $\mathcal{V}_1 = \mathcal{V}_{k+1}$.

Definition 4. A path is a walk in which all vertices and hyperedges are distinct. A closed path is called cycle.

Lemma 3.1. *If s_i (or v_j) is alive in PG and CCN, then either it is included in a cycle in the hypergraph model $\mathcal{H}$ or there exists a path ending at s_i (or v_j) from a node that is included in a cycle in $\mathcal{H}$.*

Proof. Without losing the generality, we assume that s_i is alive for some i, but by contradiction assume that neither it is included in a cycle nor there is a path from a cycle to it in the $\mathcal{H}$. Since s_i is alive there is a hyperedge $\mathcal{E}_j$ such that $\mathcal{E}_j = (\mathcal{E}_1^j, \{s_i\})$ and $\mathcal{E}_1^j = \{r_{j_1}, r_{j_2}, \cdots, r_{j_k}, e\}$ in which e is an interdependency edge between these routers and s_i and also all $r_{j_1}, r_{j_2}, \cdots, r_{j_k}$ must be alive. Therefore anyone has at least one input hyperedge in the hypergraph model. Because of early assumption none of them are not included in no cycle and there is no path from a cycle to them. By following this process we must reach some s_l or r_l that has no input hyperedge. In other word we have a path from s_l or r_l to s_i. Now since s_l (or r_l) hasn't input hyperedge, therefore it can't be alive. Hence by cascading failures on this path s_i also can't be alive and it's contradiction. □

Theorem 3.1. *The minimum number of failures that lead to total failure in PG and CCN equals to minimum number of vertices whose removals break all cycles of the hypergraph model.*

Proof. We consider that the existence of the cycle $v_1, \mathcal{E}_1, \cdots, v_i, \mathcal{E}_i, v_{i+1}, \cdots, v_{k+1}$ is equivalent with operating of all nodes $v_1, \cdots, v_{k+1}$ in PG and CCN. Because if one of them, for example v_i, isn't operating then v_i can't be active in interdependency edge $\mathcal{E}_1^i \rightarrow \{v_{i+1}\}$. Therefore this interdependency edge fails and then $\mathcal{E}_i = (\mathcal{E}_1^i, \{v_{i+1}\})$ also fails. Thus the cycle fails. Hence the existence of a cycle in our hypergraph model follows the existence of alive nodes in PG and CCN. Conversely by lemma(3.1) the existence of an alive node follows the existence of a cycle. Therefore for total failure in PG and CCN, we must break all cycles in the hypergraph. Thus we proved the theorem. □

Thus in fact we want to break all cycles of the hypergraph. A heuristics method that gives solution of this problem is that finding vertex with sharing among maximum number of cycles, removing this vertex, updating hypergraph and repeating until no cycle remains. But finding cycles passing a vertex is difficult. Hence we propose an approximate method that solve this problem by using of the number of closed walks passing a vertex. Before presenting the algorithm, first we study the adjacency matrix of a directed hypergraph and then by using of spectral properties of hypergraphs we determine the number of closed walks passing a vertex.

3.1. The number of closed walks passing a vertex

We start with some definitions in a directed hypergraph.

Definition 5. The incidence matrix of a directed hypergraph $\mathcal{H} = (\mathcal{V}_{\mathcal{H}}, \mathcal{E}_{\mathcal{H}})$ is $\mathbf{E} = [e_{ij}]_{|\mathcal{V}_{\mathcal{H}}| \times |\mathcal{E}_{\mathcal{H}}|}$ where for each $\mathcal{E}_j = (\mathcal{E}_1^j, \mathcal{E}_2^j) \in \mathcal{E}_{\mathcal{H}}$ we have:

$$e_{ij} = \begin{cases} 1 & v_i \in \mathcal{E}_1^j \\ -1 & v_i \in \mathcal{E}_2^j \\ 0 & otherwise \end{cases}$$

Definition 6. The adjacency matrix of a directed hypergraph $\mathcal{H} = (\mathcal{V}_{\mathcal{H}}, \mathcal{E}_{\mathcal{H}})$ is $\mathbf{A} = [a_{ij}]_{|\mathcal{V}_{\mathcal{H}}| \times |\mathcal{V}_{\mathcal{H}}|}$ in which:

$$\begin{array}{ll} a_{ij} = |\{\mathcal{E}_k \in \mathcal{E}_{\mathcal{H}} |\ \mathcal{E}_k = (\mathcal{E}_1^k, \mathcal{E}_2^k)\ ,\ v_i \in \mathcal{E}_1^k\ ,\ v_j \in \mathcal{E}_2^k\ \}| & if\ \ i \neq j \\ a_{ij} = 0 & if\ \ i = j \end{array}$$

(Similarly in [6] about undirected hypergraph.)

In other word a_{ij} is the number of hyperedges that v_i is a member of their bottom and v_j is a member of their tail. However this matrix don't reflex all information about a hypergraph and for having all details must use incidence matrix of a hypergraph.

Now we have the following theorem:

Theorem 3.2. *The number of directed walks of length k in the directed hypergraph $\mathcal{H} = (\mathcal{V}_{\mathcal{H}}, \mathcal{E}_{\mathcal{H}})$ from vertex v_i to vertex v_j is*

$$\mathbf{N}_k(i,j) = (\mathbf{A}^k)_{ij}$$

where $\mathbf{A}$ is the adjacency matrix of the $\mathcal{H}$ in definition (6).

Proof. By induction, let $k = 1$. It is trivial that $\mathbf{N}_1(i,j) = (\mathbf{A})_{ij}$ i.e. the number of hyperedges from v_i to v_j. Now suppose that $k = n$ and $\mathbf{N}_n(i,j) = (\mathbf{A}^n)_{ij}$, we show that $\mathbf{N}_{n+1}(i,j) = (\mathbf{A}^{n+1})_{ij}$. By principle of multiplication we know:

$$\mathbf{N}_{n+1}(i,j) = \sum_{l=1}^{|\mathcal{V}_{\mathcal{H}}|} \mathbf{N}_n(i,l)\mathbf{N}_1(l,j)$$

On the other hand by induction hypothesis, we have $\mathbf{N}_n(i,j) = (\mathbf{A}^n)_{ij}$ and $\mathbf{N}_1(i,j) = (\mathbf{A})_{ij}$. Therefore we have:

$$\mathbf{N}_{n+1}(i,j) = \sum_{l=1}^{|\mathcal{V}_{\mathcal{H}}|} \mathbf{N}_n(i,l)\mathbf{N}_1(l,j) = \sum_{l=1}^{|\mathcal{V}_{\mathcal{H}}|} (\mathbf{A}^n)_{il}\mathbf{A}_{lj} = (\mathbf{A}^{n+1})_{ij}$$

□

Now similar [6] and with slightly changes, we define the centrality of the vertex v_i, $C(i)$, as the sum of the closed directed walks of different lengths in the hypergraph starting and ending at vertex v_i i.e. $C(i) = \sum_{k=1}^{|\mathcal{E}_{\mathcal{H}}|} \frac{\mathbf{N}_k(i)}{k}$. It is trivial that there exists at least one directed cycle passing v_i for each closed directed walk starting and ending at vertex v_i. Therefore we can use $C(i)$ as a criterion for comparing the number of directed cycles passing nodes.

www.ejgta.org

4. The Approximation Algorithm

We propose an approximation algorithm that find minimum number of failures for total failure, TF algorithm. This algorithm is as follows.

TF Algorithm

Input : E, Incidence matrix of the hypergraph model correspond to the PG and CCN

Output : $Selected$, Nodes or interdependency edges that must be fail for total failure in PG and CCN

n ←size(E,1) and $Selected \leftarrow$ (0)
(E,$Selected$) ← $Cycle$(E,$Selected$)
if E ≠0
begin
    A ← $Transform$(E) and t ← $true$
    while t
    begin
        sum ← $Nwalks$(A)
        index ← indexmax(sum)
        $Selected$(index) ← 1
        E ← $Updated$(E,index)
        if E=(0) then t ← $false$
        else A ← $Transform$(E)
    end
end
return $Selected$

This algorithm consists of four functions: *Cycle Transform, Nwalks* and *Updated*.

Cycle is a function that find cycles of form $s_i\ \mathcal{E}_k\ r_j\ \mathcal{E}_l\ s_i$ in the hypergraph model where $\mathcal{E}_k = (\{s_i, e_{ij}\}, r_j)$ and $\mathcal{E}_l = (\{r_j, e_{ji}\}, s_i)$ in which e_{ij} is the interdependency edge from s_i to r_j and e_{ji} is the interdependency edge from r_j to s_i. This cycle is shown in figure (2).

Correspond to each cycle, this function removes the vertex with maximum centrality and then updates incidence matrix of the hypergraph with function $Updated$. Note that these cycles are the most robust subhypergraphs and corresponding nodes, s_i and r_j will be alive in PG and CCN as long as the cycle $s_i\ \mathcal{E}_k\ r_j\ \mathcal{E}_l\ s_i$ exists in the hypergraph model. Therefore we must first remove such cycles.

Nwalks obtains centrality of each vertex of the hypergraph. Then the algorithm choose the vertex with maximum centrality that present in maximum number of cycles approximately and removes this vertex from the hypergraph and afterward updates the incidence matrix by $Updated$

www.ejgta.org

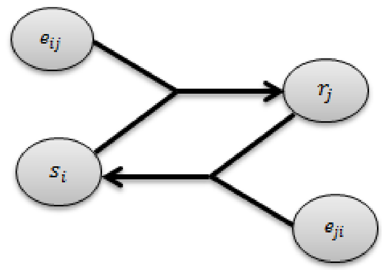

Figure 2. The robust cycle of length two

function. To find centrality of each vertex we need the adjacency matrix of $\mathcal{H}$ that we do it by using $Transform$ function. These steps repeat until matrix $\mathbf{E} \neq 0$.

Note that all of these functions have simple structure and determine their outputs in polynomial time.

5. Evaluation

We evaluate our methods extensively on numerous synthetic random directed bipartite hypergraphs in form of definition(1). In [1] was proposed a recursive model for generating realistic graphs by using random typing. Now we use it with slightly change, for generating random directed bipartite hypergraphs in which we use the two-dimensional keyboard, we select some letters from first dimension randomly and then one letter from second dimension or Vice versa, instead of one letter from first dimension and one letter from second dimension and so we introduce a hyperedge in desired form. As said in [1], they introduced an imbalance factor β that determine the modularity of generated graphs such that the modularity increases with smaller β and without any imbalance, $\beta = 1$ no significant modularity exists. This case is happen in "unipartite" graphs while we use this method for generating directed "bipartite" hypergraphs. In this case the increasing of β cause to generate more cycles of length two (Figure 2) and hence more robustness. Figure 3 shows the number of such cycles as robust cycles in generated bipartite hypergraphs with different βs.

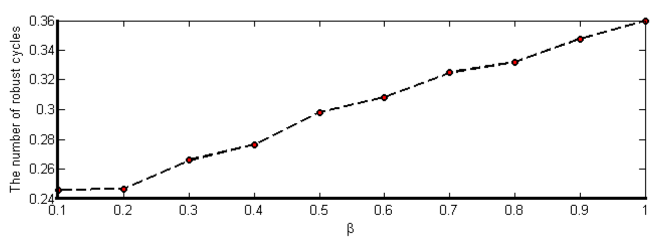

Figure 3. **The number of robust cycles vs. β of 25000 random directed bipartite hypergraph.**

We apply our method on 5000 generated directed bipartite hypergraphs for each β and then the following result is determined. As seen in Figure 4 by increasing β increases robustness.

www.ejgta.org

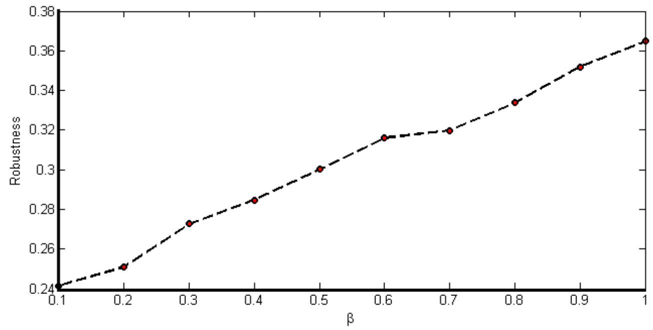

Figure 4. **Robustness vs. β of more than 5000 random directed bipartite hypergraph.**

6. Conclusion

In this paper we try to represent complex interdependencies in an interdependent multi-layered network with a directed hypergraph model. These dependencies are educed from conditional propositions that are of two kinds, conjunctive dependencies and disjunctive dependencies. Now for finding minimum number of failures that cause total failure we propose a polynomial algorithm that obtain an approximated solution for this problem.

References

[1] L. Akoglu and C. Faloutsos, RTG, A recursive realistic graph generator using random typing, *Data Min. Knowl. Discov.* **19** (2009), 194–209.

[2] C. Berge, *Graphs and Hypergraphs*, *(Elsevier, New York, 1973)*.

[3] C. Berge, *Hypergraphs, The Theory of Finite Sets*, *(North-Holland, Amsterdam, 1989)*.

[4] A. Bernstein, D. Bienstock, D. Hay, M. Uzunoglu and G. Zussman, Power grid vulnerability to geographically correlated failures-analysis and control implications, *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, DOI: 10.1109/INFOCOM.2014.6848211.

[5] S. V. Buldyrev, R. Parshani, G. Paul, H. E. Stanley and S. Havlin, Catastrophic cascade of failures in interdependent networks, *Nature* **464** (2010), 1025–1028.

[6] E. Estrada and J. Rodriguez-Velazquez, Complex networks as hypergraphs, *Arxiv preprint physics/0505137* (2005).

[7] J. Gao, S. V. Buldyrev, H. E. Stanley and S. Havlin, Networks formed from interdependent networks, *Nature Physics* **8** (2011), 40–48.

[8] C. MR, M. K. Menon and A. Vijayakumar, Some diameter notions in lexicographic product, *Electron. J. Graph Theory Appl.* **6** (2) (2018), 258–268

[9] D. T. Nguyen, Y. Shen, and M. T. Thai, Detecting critical nodes in interdependent power networks for vulnerability assessment, *IEEE Transactions on Smart Grid* **4** (2013), 151–159.

[10] M. Parandehgheibi and E. Modiano, Robustness of interdependent networks: The case of communication networks and the power grid, *Global Conference (GLOBECOM), IEEE*, (2013), 2164–2169.

[11] S. Rinaldi, J. Peerenboom and T. Kelly, Identifying, understanding and analyzing critical infrastructure interdependencies,*Control Systems, IEEE*, **21** (2001),11–25.

[12] V. Rosato, L. Issacharoff, F. Tiriticco, S. Meloni, S. Porcellinis and R. Setola, Modelling interdependent infrastructures using interacting dynamical models, *International Journal of Critical Infrastructures* **4** (2008), 63–79.

[13] A. Sen, A. Mazumder, J. Banerjee, A. Das and R. Compton, Identification of K Most Vulnerable Nodes in Multi-layered Network Using a New Model of Interdependency, *Computer Communications Workshops (INFOCOM WKSHPS), IEEE Conference on*, (2014), 831–836.

[14] J. Shao, S. V. Buldyrev, S. Havlin, and H. E. Stanley, Cascade of failures in coupled network systems with multiple support-dependence relations, *Physical Review E* **83** (2011).

[15] P. Shetty D. and M. P. Lakshmi, Minimizing the maximum sender interference by deploying additional nodes in a wireless sensor network, *Electron. J. Graph Theory Appl.* **7** (1) (2019), 169–182.

[16] P. Zhang, S. Peeta and T. Friesz, Dynamic game theoretic model of multi-layer infrastructure networks, *Networks and Spatial Economics* **5** (2005), 147–178.